

Распараллеливание Data-Intensive приложений с помощью библиотеки DISLIB на десятки тысяч ядер

Антон Корж

Т-Платформы

Member of Graph500 Steering Committee

What is the Graph500?

- New benchmark to complement the Top 500 for large-scale data analysis problems
- International Multidisciplinary Steering Committee
 - Jim Ang, David A. Bader, Brian Barrett, Jon Berry, Bill Brantley, Almadena Chtchelkanova, John Daly, John Feo, Michael Garland, John Gilbert, Bill Gropp, Bill Harrod, Bruce Hendrickson, Anton Korzh, Jure Leskovec, Bob Lucas, Andrew Lumsdaine, Mike Merrill, Hans Meuer, David Mizell, Shoaib Mufti, Richard Murphy, Nick Nystrom, Fabrizio Petrini, Wilf Pinfold, Steve Poole, Arun Rodrigues, Rob Schreiber, John Simmons, Marc Snir, Thomas Sterling, Blair Sullivan, T.C. Tuan, Jeff Vetter, Mike Vildibill
- Three Kernels
 - Search (Concurrent Search, the Ranking Kernel)
 - Optimization (Single Source Shortest Path, almost released)
 - Edge Oriented (Maximal Independent Set, in specification)

History of the Graph500

- Graph500 announced at ISC10 (June 2010)
- 1st Graph500 List: 9 machines at SC10 (Nov. 2010)
- 2nd Graph500 List: 29 machines at ISC11 (June 2011)
- 3rd Graph500 List: 51 machines at SC11 (Nov. 2011)
- 4th Graph500 List: 88 entries at ISC 12 (June 2012)
- 5th Graph500 List: 124 entries at SC12 (Nov. 2012)
- 6th Graph500 List: 142 entries at ISC13 (June 2013)
- 7th Graph500 List: 160 entries at SC13 (Nov. 2013) [TODAY!]

Five Business Areas

- Cybersecurity
 - 15 Billion Log Entries/Day (for large enterprises)
 - Full Data Scan with End-to-End Join Required
- Medical Informatics
 - 50M patient records, 20-200 records/patient, billions of individuals
 - Entity Resolution Important
- Social Networks
 - Example, Facebook, Twitter
 - Nearly Unbounded Dataset Size
- Data Enrichment
 - Easily PB of data
 - Example: Maritime Domain Awareness
 - Hundreds of Millions of Transponders
 - Tens of Thousands of Cargo Ships
 - Tens of Millions of Pieces of Bulk Cargo
 - May involve additional data (images, etc.)
- Symbolic Networks
 - Example, the Human Brain
 - 25B Neurons
 - 7,000+ Connections/Neuron

[Home](#)[Complete Results](#)[Benchmarks](#)[Log In](#)

The Graph 500 List

Top 10 (June 2012)

Rank	Machine
1	DOE/SC/Argonne National Laboratory - Mira/BlueGene/Q (32768 nodes, 524288 cores)
1	LLNL - Sequoia/Blue Gene/Q (32768 nodes, 524288 cores)
2	DARPA Trial Subset, IBM Development Engineering - Power 775, POWER7 8C 3.836 GHz (1024 nodes, 32768 cores)
3	Information Technology Center, The University of Tokyo - Oakleaf-FX (Fujitsu PRIMEHPC FX 10) (4800 nodes, 76800 cores)
4	GSIC Center, Tokyo Institute of Technology - HP Cluster Platform SL390s G7 (three Tesla cards per node) (1366 nodes, 16392 cores)
5	Brookhaven National Laboratory - BLUE GENE/Q (1024 nodes, 16384 cores)
6	DOE/SC/Argonne National Laboratory - Vesta/BlueGene/Q (1024 nodes, 16384 cores)

Brief Introduction

Data intensive supercomputer applications are increasingly important for HPC workloads, but are ill-suited for platforms designed for 3D physics simulations. Current benchmarks and performance metrics do not provide useful information on the suitability of supercomputing systems for data intensive applications. A new set of benchmarks is needed in order to guide the design of hardware architectures and software systems intended to support such applications and to help procurements. Graph algorithms are a core part of many analytics workloads.

Backed by a [steering committee](#) of over 50 international HPC experts from academia, industry, and national laboratories, Graph 500 will establish a set of large-scale benchmarks for these applications. The Graph 500 steering committee is in the process of developing comprehensive benchmarks to address three application kernels: concurrent search, optimization (single source shortest path), and edge-oriented (maximal independent set). Further, we are in the process of addressing five graph-related business areas: Cybersecurity, Medical Informatics, Data Enrichment, Social Networks, and Symbolic Networks.

This is the first serious approach to complement the Top 500 with data intensive applications. Additionally, we are working with the SPEC committee to include our benchmark in their CPU benchmark suite. We anticipate the list will rotate between ISC and SC in future years.

The Graph 500 was announced at ISC2010 and the first list appeared at SC2010.

Submissions

November 2012 List

The submission deadline for the November list is October 22, 2012.

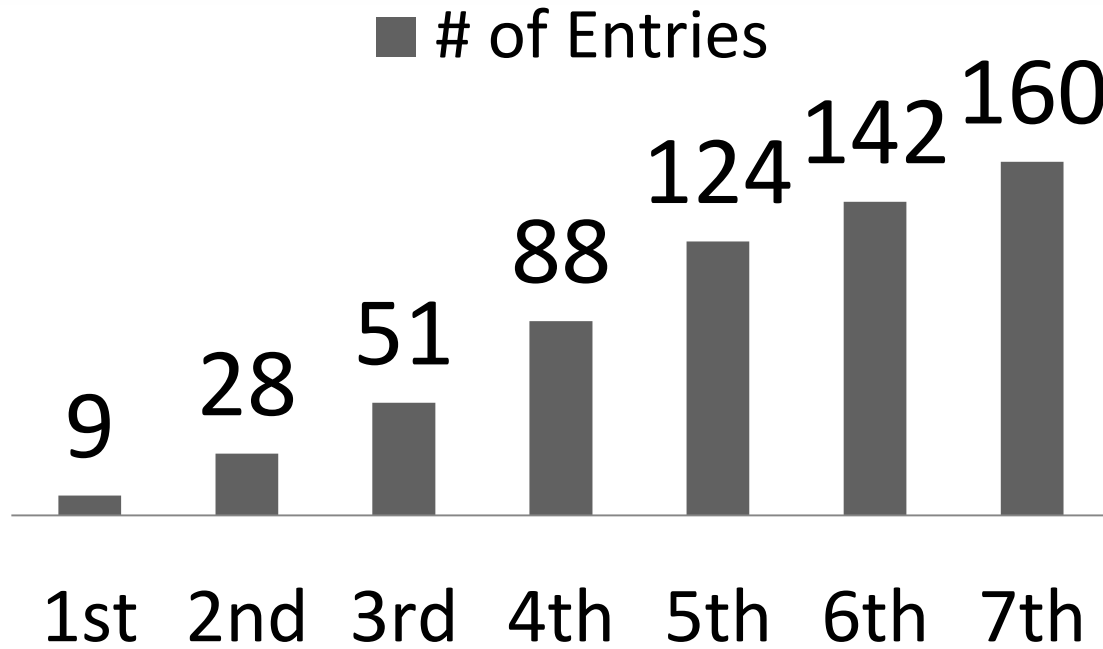
You must be logged in to access the submission form. If you do not have an account, [request one here](#).

If you already logged in you can reach the [submission form here](#)

The following information will be collected:

1. Computer Information:
 - Computer/System Name
 - Manufacturer
 - Computer Type/Model
 - Installation Site
 - Location
 - Year of Installation/Last Major Upgrade
 - Field of Use: government, university, industry, etc.
 - Field of Application: geophysics, automotive, etc.
 - Number of Nodes
 - Number of Cores
 - Main Memory Size
 - Total system power
 - Interconnection network
 - Graph 500 Implementation Used: reference, custom, etc.
 - Contact Name
 - Contact Email
2. Benchmark Information:
 - Problem Scale
 - GigaTEPS
 - Graph Construction Time
 - Full Benchmark Result

7th Graph 500 List (followed by special highlights)



7th Graph 500 List

Country	# entries	% entries
Amsterdam	2	1.3%
Australia	1	0.6%
Canada	3	1.9%
China	6	3.8%
France	2	1.3%
Germany	3	1.9%
Italy	2	1.3%
Japan	39	24.4%
Luxembourg	1	0.6%
Poland	1	0.6%
Russia	6	3.8%
Russian Federation	1	0.6%
South Korea	1	0.6%
Switzerland	6	3.8%
Taiwan	6	3.8%
UK	4	2.5%
USA	76	47.5%
Grand Total	160	

Lawrence Livermore National Laboratory's
DOE/NNSA/LLNL Sequoia

is ranked

No.1

on the Graph500 Ranking of Supercomputers with
15363 GE/s on Scale 40
on the 7th Graph500 list published at the International
Supercomputing Conference, November 19, 2013.

Congratulations from the Graph500 Executive Committee



David A. Bader

David A. Bader

Andrew Lumsdaine

Andrew Lumsdaine

Richard C. Murphy

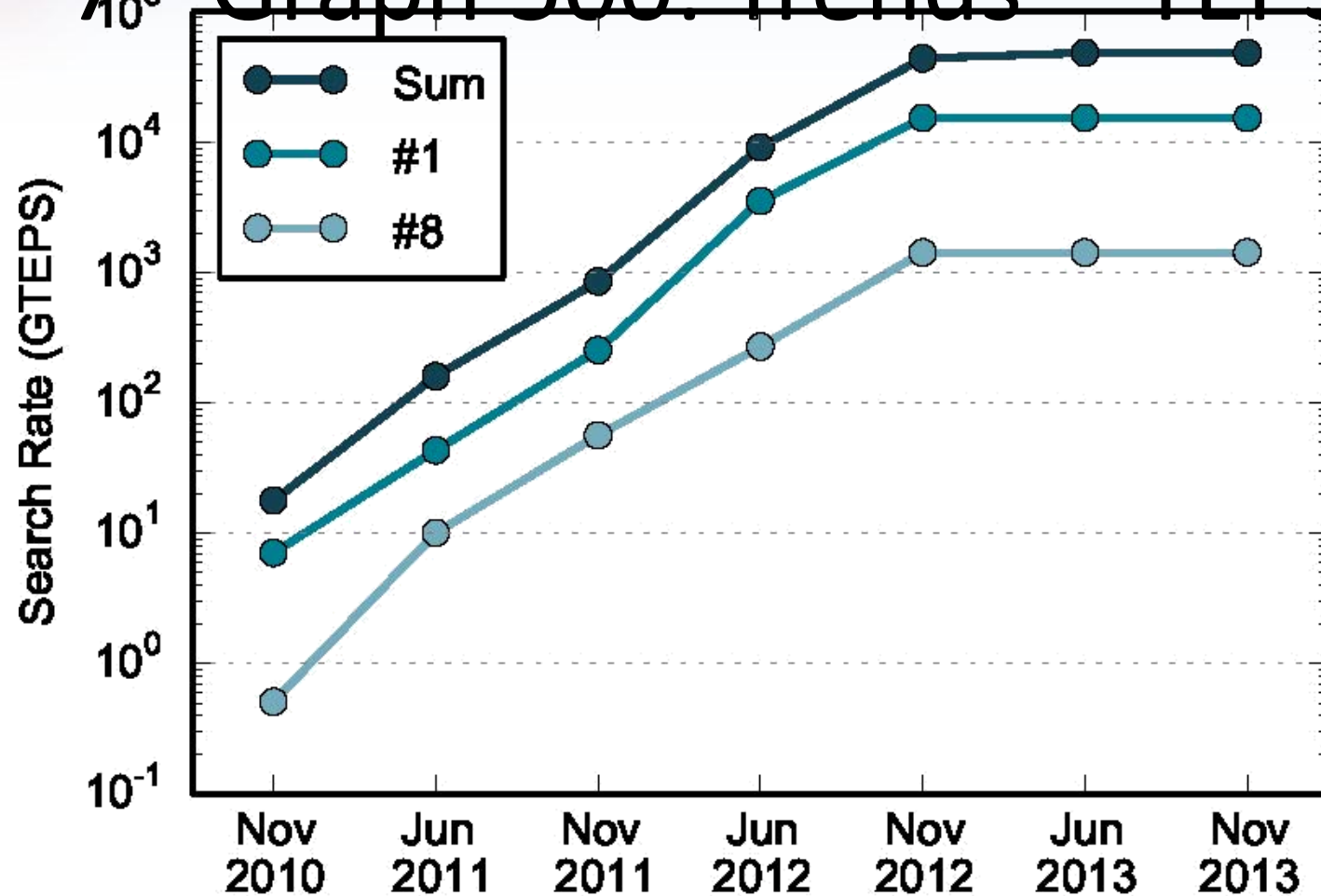
Richard Murphy

Marc Snir

Marc Snir

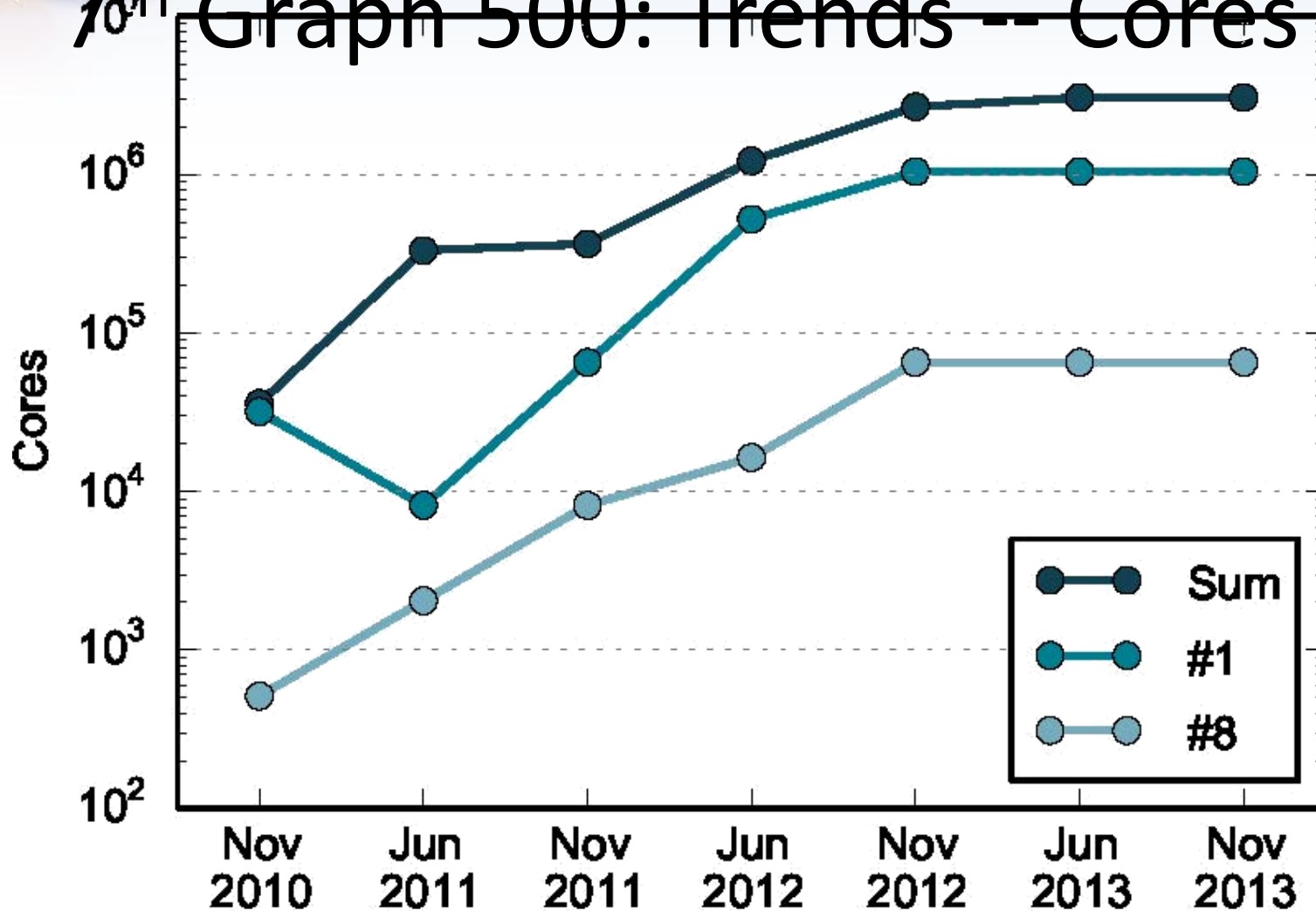
Graph500 Executive Committee

7th Graph 500: Trends -- TEPS



Slide credit: Scott Beamer

7th Graph 500: Trends -- Cores

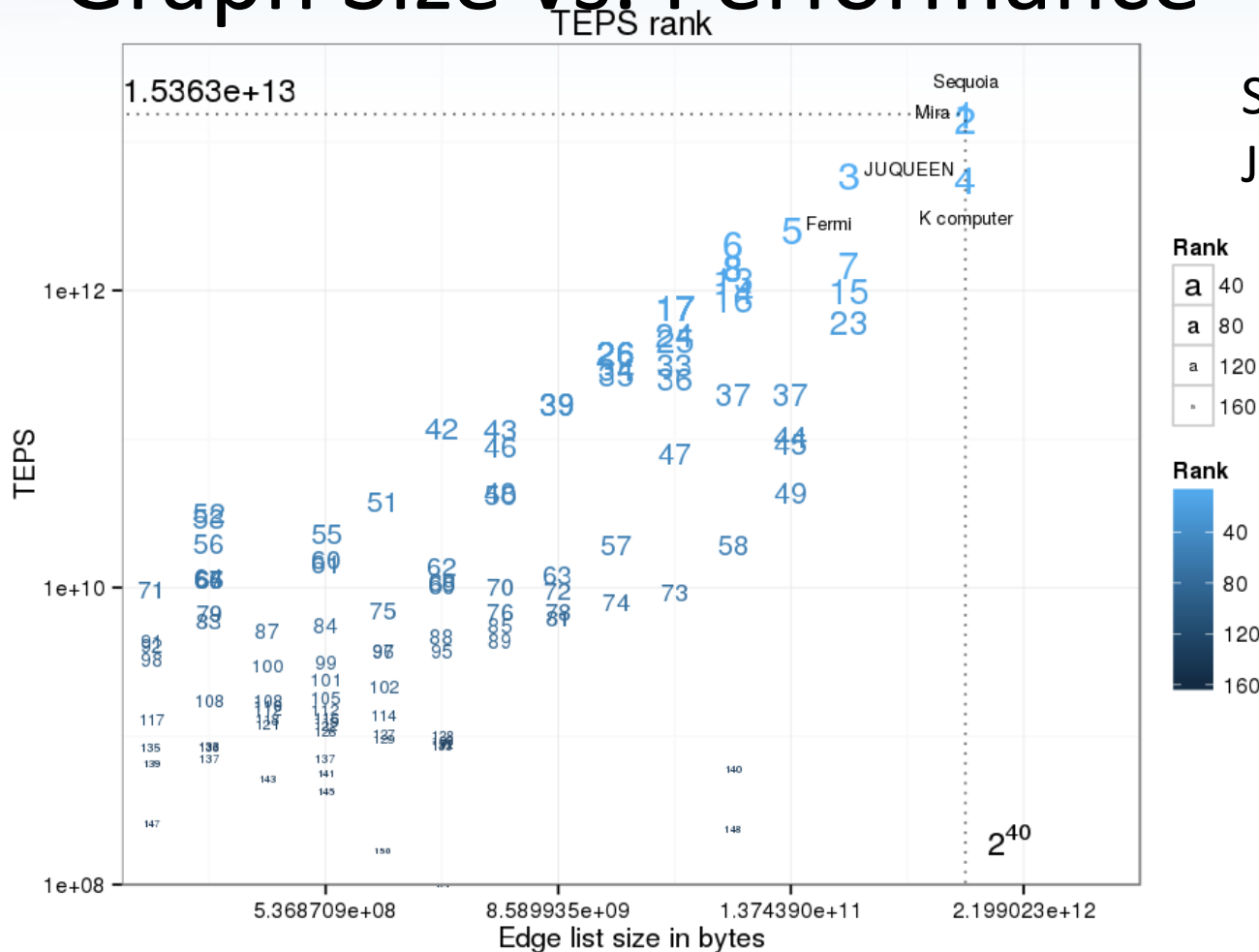


Slide credit: Scott Beamer

Performance (Edges/Second), (TEPS)

7th Graph500 List

Graph Size vs. Performance



Slide credit:
Jason Riedy

Normalized Graph Data
Structure Size

Highlights of the 7th Graph500 List

- The list is growing!
- Top systems have leveled off
- Three vendors account for approximately half the list.
- Graph500 and Top500 rankings are not strongly correlated!
 - Top500's #1 system (Tianhe-2) is ranked #6 on Graph500
 - Graph500's #1 system (Sequoia) is ranked #3 on Top500

DISLIB

- Расширение SHMEM активными сообщениями
- Вместо `shmem_put` → `shmem_send`
- Прозрачная агрегация сообщений
- Эффективная реализация для кластеров с малореактивным интерконнектом
- Поддержка многоядерности

DISLIB History of Success

- 2009 NPB UA, dcmf version (BlueGene/P)
- 2010 GASNET-version (IB)
- 2011 Graph500 (BFS)
- 2011 MPI version +multicore optimized
- 2013 Quantum Computer
- 2014 Students, SSSP

```
#include "dislib.h"

int *data;

void allgather_hndl(int from, void* message, int size)
{ data[from] = * (int*)message; }

void main(int argc, char** argv) {
    shmem_init(&argc,&argv);
    shmem_register_handler(allgather_hndl,1);
    data=malloc( sizeof(int) * num_pes() );
    data[my_pe()] = 57*my_pe();
    shmem_barrier_all();

    for(int i=0;i<num_pes();i++)
        shmem_send (data+my_pe(),1,sizeof(int),i);
    shmem_barrier_all();

    shmem_finalize();
}
```

BFS

```
if (VERTEX_OWNER(root) == my_pe()) {  
    SET_VISITED(root);  
    q1[0]=VERTEX_LOCAL(root);  
    qc=1;  
}  
shmem_register_handler(visithndl,1);  
shmem_barrier_all();  
sum=1;  
while(sum!=0) {  
    for(i=0;i<qc;i++)  
        for(j=g->rowsIndices[q1[i]];j<g->rowsIndices[q1[i]+1];j++)  
            send_vertex(g->endV[j]);  
    shmem_barrier_all();  
    qc=q2c;q2c=0;int *tmp=q1;q1=q2;q2=tmp;  
  
    sum=qc;  
    shmem_long_allsum(&sum);  
}
```


Active messages

```
void visithndl(int from, void* dat, int size) {  
    int vloc = ((int*) dat)[0];  
    if (!TEST_VISITEDLOC(vloc)) {  
        SET_VISITEDLOC(vloc);  
        q2[q2c++] = vloc;  
    }  
}
```

```
inline void send_vertex (int64_t glob) {  
    int pe = VERTEX_OWNER(glob);  
    int vloc = VERTEX_LOCAL(glob);  
    shmem_send(&vloc,1,4,pe);  
}
```

```
while(sum!=0) {
```

```
    while(sum!=0) {
```

```
        for(i=0;i<qc;i++)
```

```
            for(j=g->rowsIndices[q1[i]];j<g->rowsIndices[q1[i]+1];j++)
```

```
                if(g->weights[j]<delta)    send_relax(g->endV[j],dist[q1[i]]+g->weights[j]);
```

```
    shmem_barrier_all();
```

```
    qc=q2c;q2c=0;int *tmp=q1;q1=q2;q2=tmp;
```

```
    sum=qc;
```

```
    shmem_long_allsum(&sum);
```

```
}
```

```
for(i=0;i<nlocalverts;i++)
```

```
    if(dist[i]>=glob_mindelta && dist[i] < glob_maxdelta) {
```

```
        for(j=g->rowsIndices[i];j<g->rowsIndices[i+1];j++)
```

```
            if(g->weights[j]>=delta)    send_relax(g->endV[j],dist[i]+g->weights[j]);
```

```
    }
```

```
shmem_barrier_all();
```

```
glob_mindelta=glob_maxdelta;
```

```
glob_maxdelta+=delta;
```

```
qc=0;sum=0;
```

```
for(i=0;i<nlocalverts;i++)
```

```
    if(dist[i]>=glob_mindelta) {
```

```
        sum++;
```

```
        if (dist[i] < glob_maxdelta)    q1[qc++]=i;
```

```
    }
```

```
shmem_long_allsum(&sum);
```

```
}
```

SSSP

Delta-stepping

```
void relaxhndl(int from, void* dat, int size) {  
    double w = ((double*) dat)[0];  
    int vloc = ((int*) dat)[2];  
if (glob_dist[vloc] < 0 || glob_dist[vloc] > w) {  
        glob_dist[vloc] = w;  
        if(w < glob_maxdelta)      q2[q2c++] = vloc;  
    }  
}
```

```
void send_relax(int64_t glob, double weight) {  
    int pe = VERTEX_OWNER(glob);  
    int vloc[3];  
    double* w = (void*)vloc;  
    *w = weight;  
    vloc[2] = VERTEX_LOCAL(glob);  
    shmem_send(&vloc,2,12,pe);  
}
```

```

void askhndl(int from, void* dat, int size) {
    int vloc = ((int*) dat)[0];
    int gfrom = VERTEX_TO_GLOBAL(from,((int*) dat)[1]);

    if(glob_dist[vloc]<glob_mindelta || glob_dist[vloc] >= glob_maxdelta)
        return;

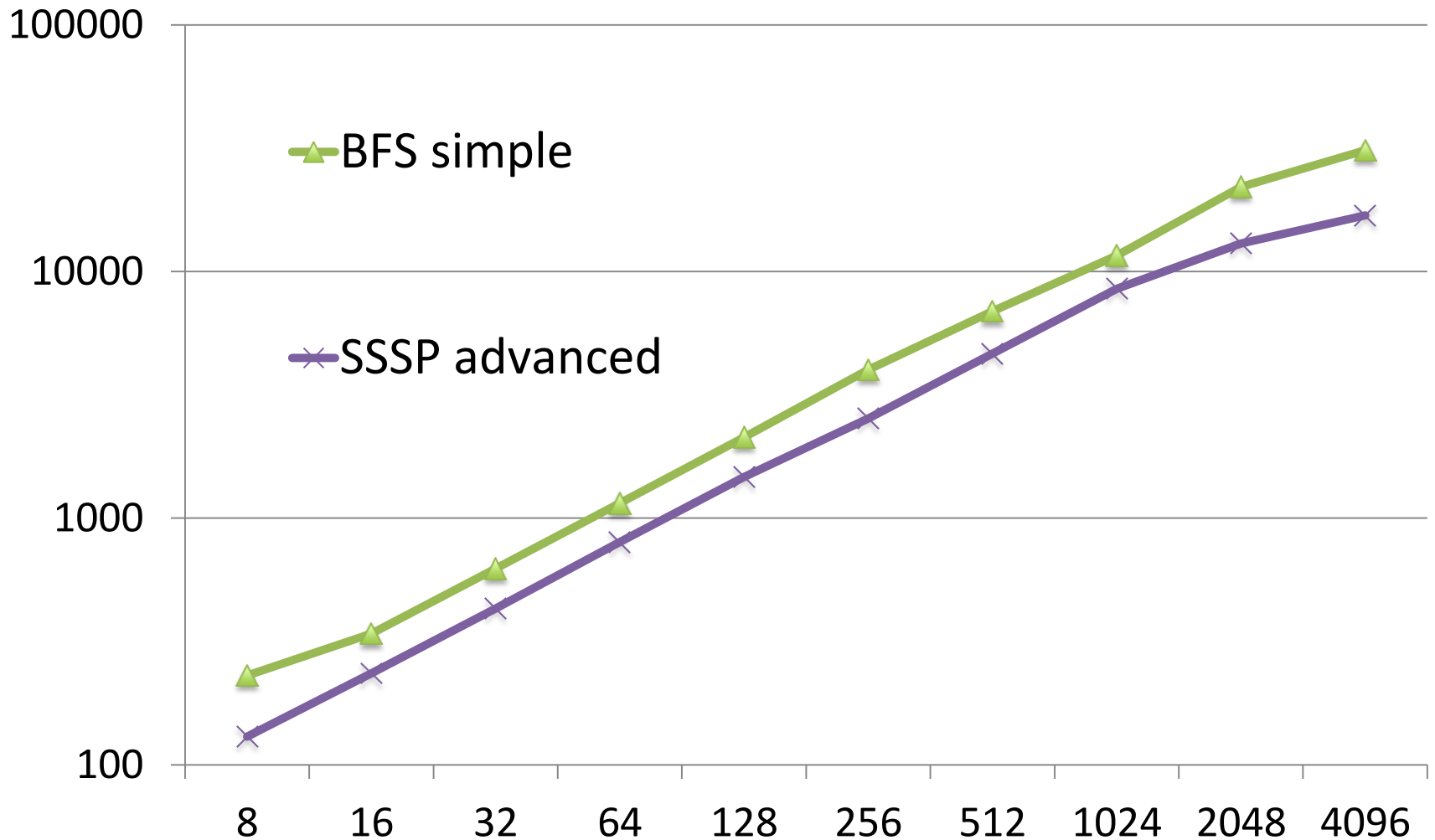
    int j;
    for(j=glob_g->rowsIndices[vloc];j<glob_g->rowsIndices[vloc+1];j++)
        if(glob_g->endV[j]==gfrom) break; //first and lightest
    double ew=glob_g->weights[j];

    if(ew<glob_delta) return;

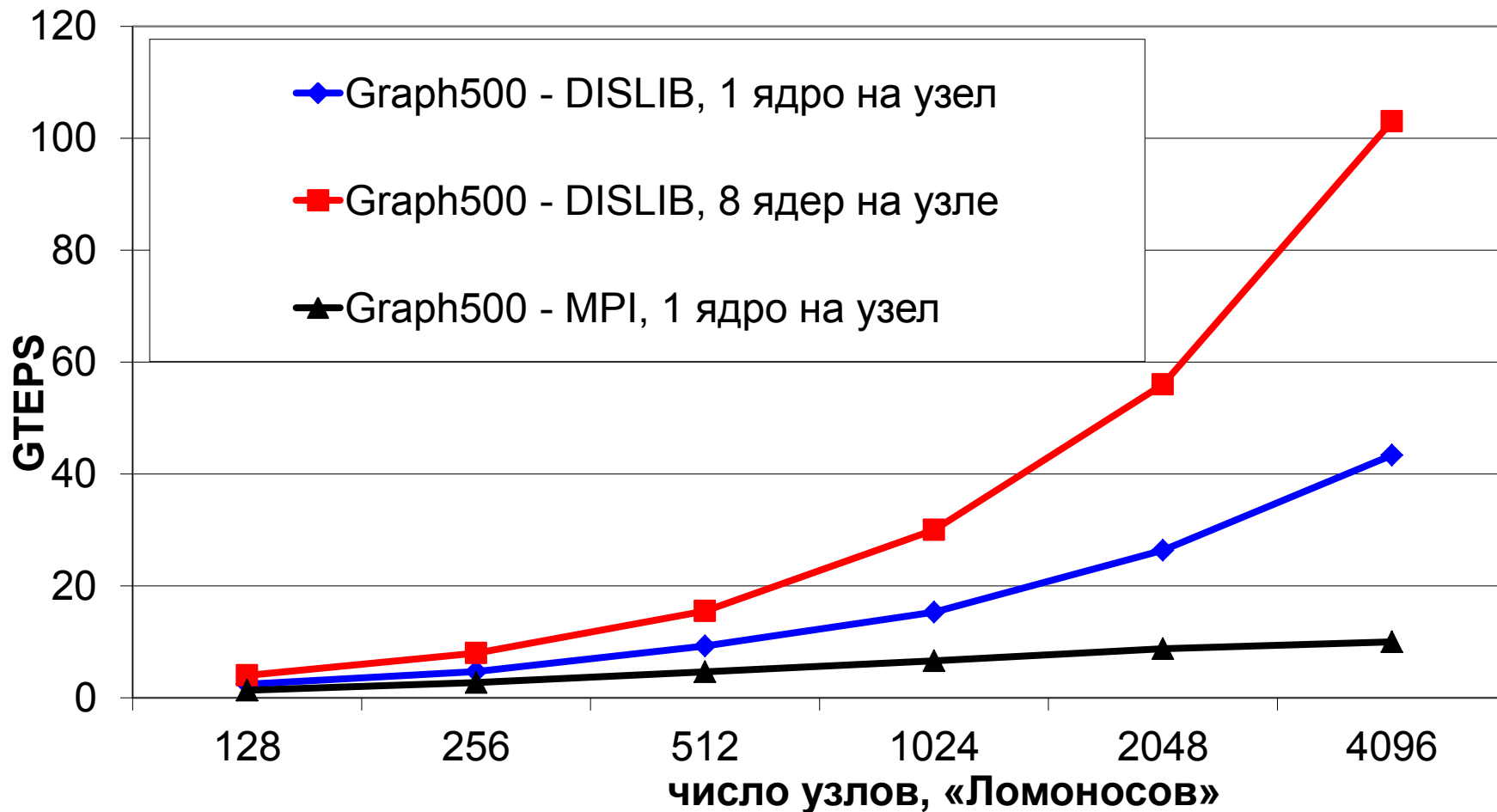
    int reply[3];
    double* ww = (void*)reply;
    *ww = glob_dist[vloc]+ew;
    reply[2] = vfrom;
    shmem_sendnb(reply,2,12,from,NULL,0);
}

```

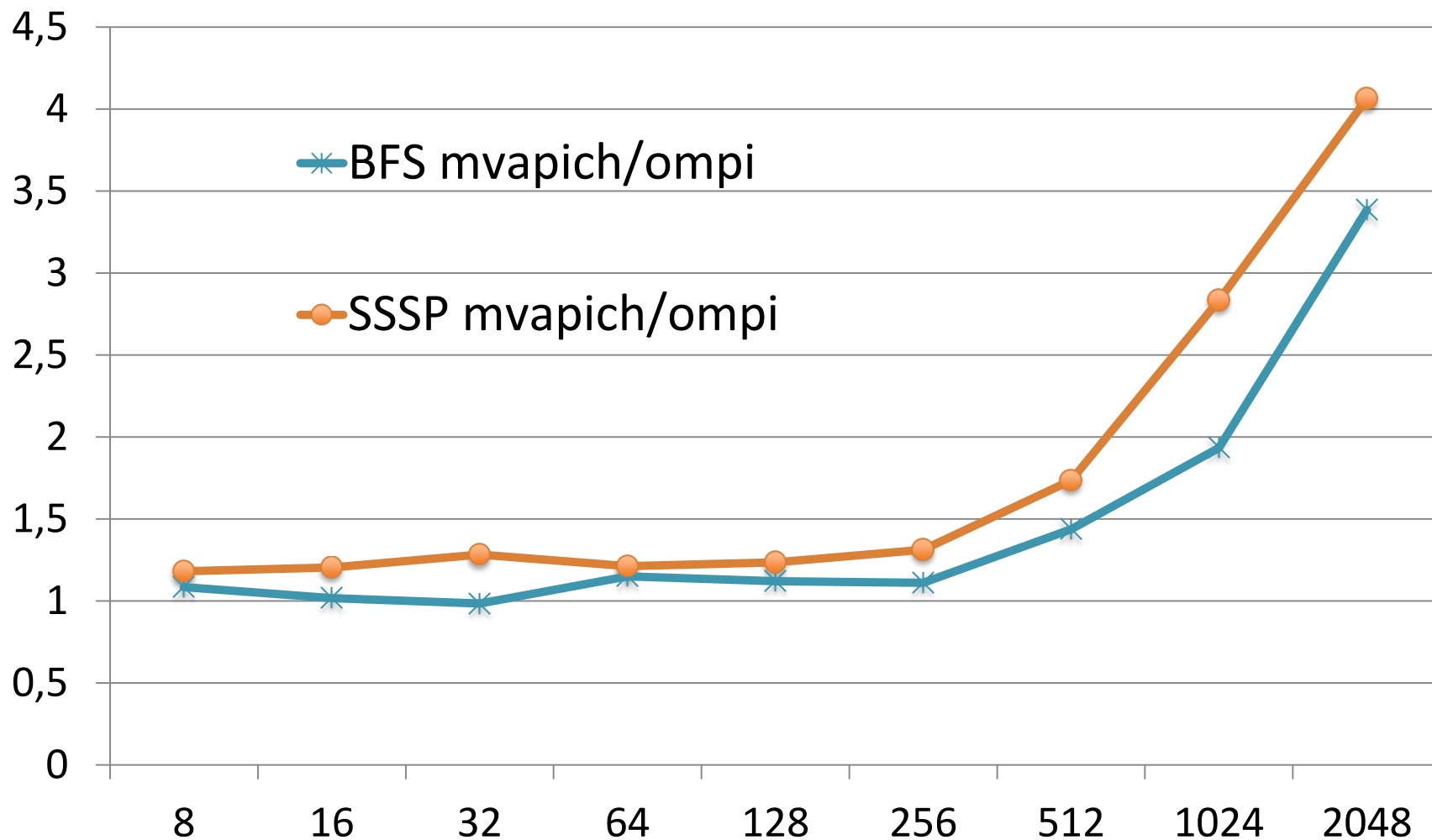
DISLIB weak scaling MTEPS/cores



Graph500 BFS, Nov/June 2011



DISLIB/MPI at scale



Try DISLIB

- Lomonosov : /opt/dislib
- /opt/dislib/graph (in few days)
- Feedback: anton@korzh.ru